Testcases
SEG2105 Summer 2020: Assignment 1
Name: Morris (Jun Yi) Cai
S/N: 3000676896

Test Cases Run on:

```
      java version "1.8.0_251"
      Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
      Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)
```

Test cases compiled on:

```
      javac 1.8.0_251
```

What still doesn't work:

- ClientConsole:
    - Displaying "Client is attempting to connect to server" before the connection is established
- ServerConsole/EchoServer:
    - Starting server with non default port (not 5555) will not allow clients to connect. Weirdly, #setport works fine and the server runs on an alternate port if started on default, then #stop, then #setport. This may also be a ClientConsole problem but I can't seem to figure it out.

<hr>
Testcase 2001: PASSED
System: Simple Chat   Phase: 2 to 4
Server startup check with default arguments
Severity: 1

Instructions:
1. At the console, enter: java EchoServer.

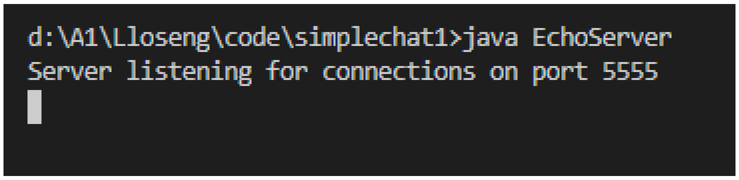
Expected result:
1. The server reports that it is listening for clients by displaying the following message:

```
Server listening for clients on port 5555
```

2. The server console awaits for user input.

Cleanup:
Hit CTRL+C to kill the server.

Console Logs



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
```

<hr>
Testcase 2002: PASSED
System: Simple Chat   Phase: 2
Client startup check without a login
Severity: 1

Instructions:
1. At the console, enter: java ClientConsole.

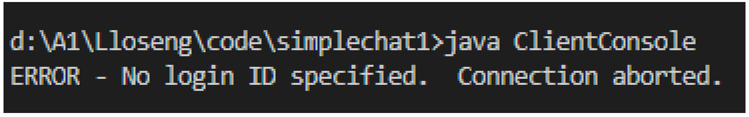
Expected result:
1. The client reports it cannot connect without a login by displaying:

ERROR - No login ID specified. Connection aborted.

2. The client terminates.

Cleanup: (if client is still active)
Hit CTRL+C to kill the client.

Console Logs



```
d:\A1\Lloseng\code\simplechat1>java ClientConsole
ERROR - No login ID specified.  Connection aborted.
```

<hr>

Testcase 2003: PASSED
System: Simple Chat   Phase: 2
Client startup check with a login and without a server
Severity: 1

Instructions:
1. At the console, enter: java ClientConsole &lt;loginID&gt; where loginID is the name you wish to be identified by.

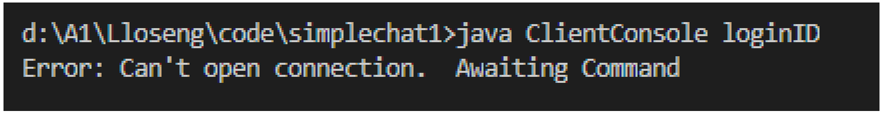
Expected result:
1. The client reports it cannot connect to a server by displaying:

Cannot open connection. Awaiting command.

2. The client waits for user input

Cleanup:
Hit CTRL+C to kill the client.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java ClientConsole loginID
Error: Can't open connection.  Awaiting Command
```

<hr>
Testcase 2004: PASSED WITH DIFFERENT PRINTOUTS
System: Simple Chat   Phase: 2
Client connection with default arguments
Severity: 1

Instructions:

1. Start a server (Testcase 2001, instruction 1)

2. On a different console on the same computer, start a client (Testcase 2003, instruction 1)

Expected results:
1. The server displays the following messages in sequence:

A new client is attempting to connect to the server.
Message received #login &lt;loginID&gt; from null.
&lt;loginID&gt; has logged on.

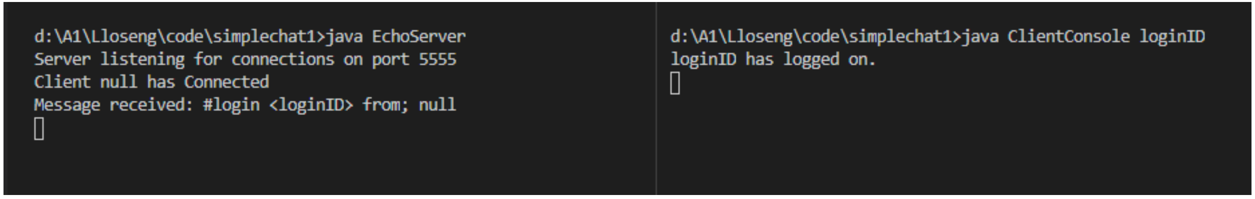
2. The client displays message:

&lt;loginID&gt; has logged on.

3. The client and the server wait for user input.

Cleanup: (unless proceeding to Testcase 2005)
Hit CTRL+C to kill the client.
Hit CTRL+C to kill the server.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <loginID> from; null
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole loginID
loginID has logged on.
```

Comments:
In the OCSF framework, there is no hook method that allows printouts for when a client is attempting to connect, but before client has connected. The first available hook that is accessed when a client is connecting is clientConnected(), which in this case should be the last printout, but will always be the first.

<hr>
Testcase 2005: PASSED
System: Simple Chat   Phase: 2
Client Data transfer and data echo
Severity: 1

Instructions:
1. Start a server and a client on the same computer and connect using default arguments (Testcase 2004 instructions).

2. Once connected, type in data on the client console and press ENTER.

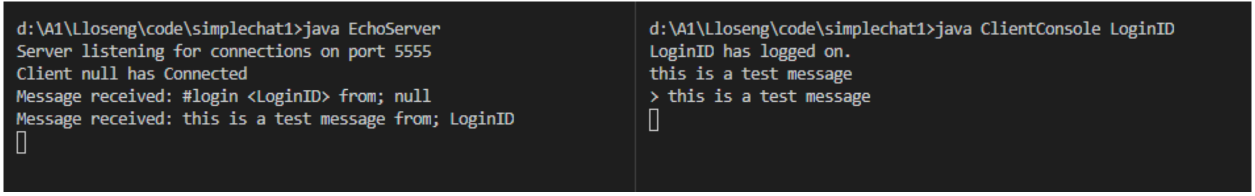
Expected results:
1. The message is echoed on the client side, but is preceded by the sender's loginID and the greater than symbol("&gt;").

2. The server displays a message similar to the following:

Message received: &lt;user input&gt; from &lt;loginID&gt;

Cleanup: (unless proceeding to Testcase 2006)
Hit CTRL+C to kill the client.
Hit CTRL+C to kill the server.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <LoginID> from; null
Message received: this is a test message from; LoginID
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole LoginID
LoginID has logged on.
this is a test message
> this is a test message
```

<hr>
Testcase 2006: PASSED
System: Simple Chat   Phase: 2
Multiple local connections
Severity: 1

Instructions:
1. Start a server and multiple clients with DIFFERENT loginIDs and connect them to the server using default arguments. (Testcase 2005 instructions).

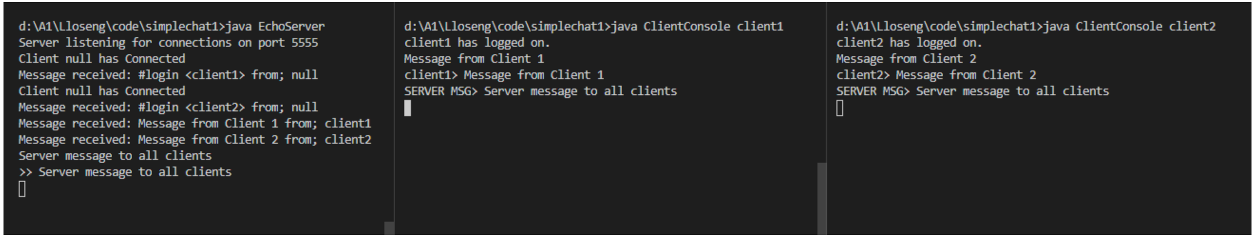
2. Start typing on all the client consoles AND the server console, pressing ENTER to send each message.

Expected results:
1. All client messages are echoed as in Testcase 2005.

2. All messages from the server console are echoed on the server console and to all clients, but are preceeded by "SERVER MESSAGE&gt; ".

Cleanup:
Hit CTRL+C to kill the clients
Hit CTRL+C to kill the servers

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client1> from; null
Client null has Connected
Message received: #login <client2> from; null
Message received: Message from Client 1 from; client1
Message received: Message from Client 2 from; client2
Server message to all clients
>> Server message to all clients
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
Message from Client 1
client1> Message from Client 1
SERVER MSG> Server message to all clients
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client2
client2 has logged on.
Message from Client 2
client2> Message from Client 2
SERVER MSG> Server message to all clients
```

<hr>
Testcase 2007: PASSED
System: Simple Chat   Phase: 2 and subsequent
Server termination command check
Severity: 1

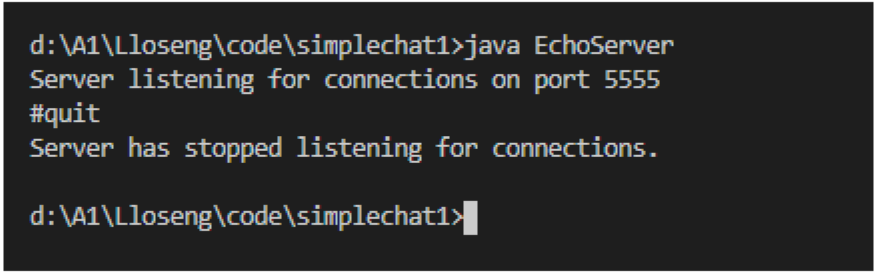
Instructions:
1. Start a server (Testcase 2001 instruction 1) using default arguments.

2. Type "#quit" into the server's console.

Expected result:
1. The server quits.

Cleanup (If the server is still active):
Hit CTRL+C to kill the server.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
#quit
Server has stopped listening for connections.

d:\A1\Lloseng\code\simplechat1>
```

<hr>
Testcase 2008 : PASSED
System: Simple Chat   Phase: 2 and 3
Server stop check
Severity: 2

Instructions:
1. Start a server (Testcase 1001, instruction 1).

2. Connect a client to the server (Testcase 2004).

3. Type "#stop" into the server's console.

4. Type in a message on the client and send it.

5. Attempt to connect other clients.

6. Restart the server.

Expected results:
1. The server displays:

Server has stopped listening for connections.

2. The client displays:

WARNING - Server has stopped listening for connections.

3. Data echoing still works normally.

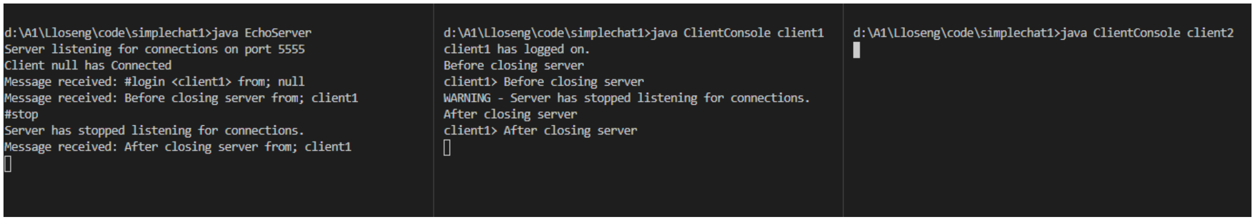
4. Attempting to connect other clients will hang the console as these new

connections are put on backlog until the max backlog (default of 10) is reached.

5. Restarting the server will cause the backlogged clients to connect properly.

Cleanup: (Unless proceeding to Testcase 2008)
Type #quit to kill the server
Hit CTRL+C to kill the client

Console Logs:
Connecting client1 > Stop listening > attempting to connect client 2

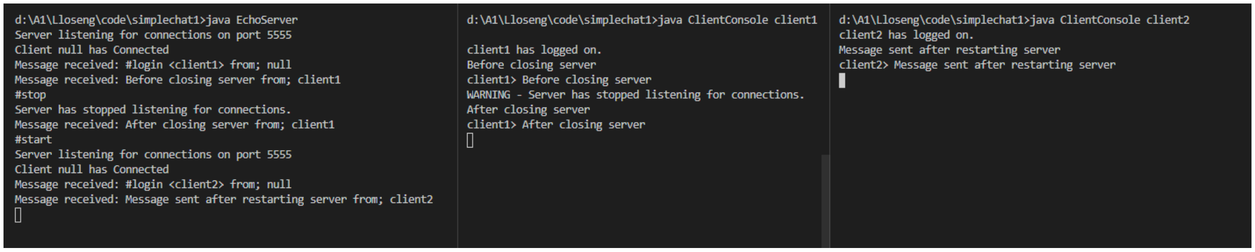


```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client1> from; null
Message received: Before closing server from; client1
#stop
Server has stopped listening for connections.
Message received: After closing server from; client1
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
Before closing server
client1> Before closing server
WARNING - Server has stopped listening for connections.
After closing server
client1> After closing server
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client2
```

After restarting server, client 2 reconnects



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client1> from; null
Message received: Before closing server from; client1
#stop
Server has stopped listening for connections.
Message received: After closing server from; client1
#start
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client2> from; null
Message received: Message sent after restarting server from; client2
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1

client1 has logged on.
Before closing server
client1> Before closing server
WARNING - Server has stopped listening for connections.
After closing server
client1> After closing server
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client2
client2 has logged on.
Message sent after restarting server
client2> Message sent after restarting server
```

<hr>
Testcase 2009: PASSED
System: Simple Chat   Phase: 2 and 3
Server close command check
Severity: 2

Instructions:
1. Start a server and connect a client to it. (Testcase 2007 instruction 1)

2. Stop the server using the #stop command.

3. Type "#close" into the server's console.

Expected result:
1. Server displays in sequence:

Server has stopped listening for connections.

&lt;loginID&gt; has disconnected.

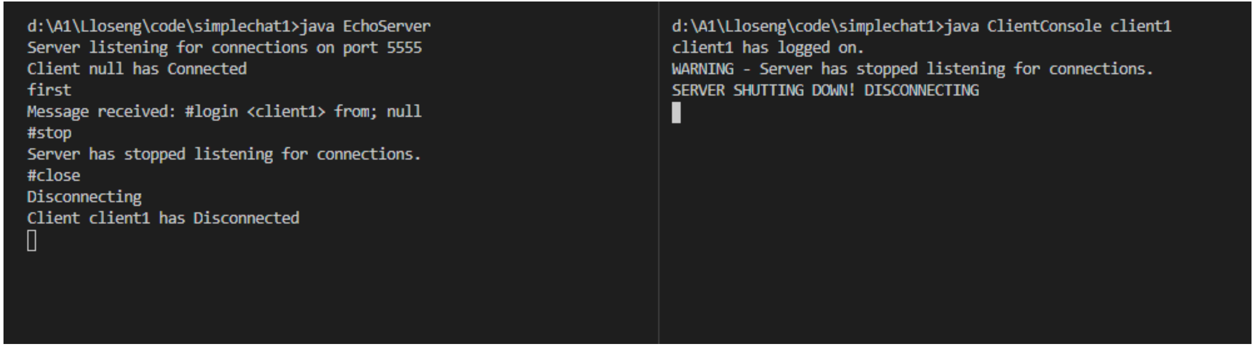
2. The client displays:

```
WARNING - The server has stopped listening for connections
SERVER SHUTTING DOWN! DISCONNECTING!
Abnormal termination of connection.
```

and then waits for input.

Cleanup: (Unless proceeding to Testcase 2009)
Hit CTRL+C to kill the client.
Hit CTRL+C to kill the server.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
first
Message received: #login <client1> from; null
#stop
Server has stopped listening for connections.
#close
Disconnecting
Client client1 has Disconnected
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
WARNING - Server has stopped listening for connections.
SERVER SHUTTING DOWN! DISCONNECTING
```

<hr>
Testcase 2009: PASSED
System: Simple Chat   Phase: 2 and subsequent
Server restart
Severity: 1

Instructions:
1. Start a server, connect a client, and close the server. (Testcase 2008)

2. Type "#start" into the server's console.

3. Attempt to connect a client.

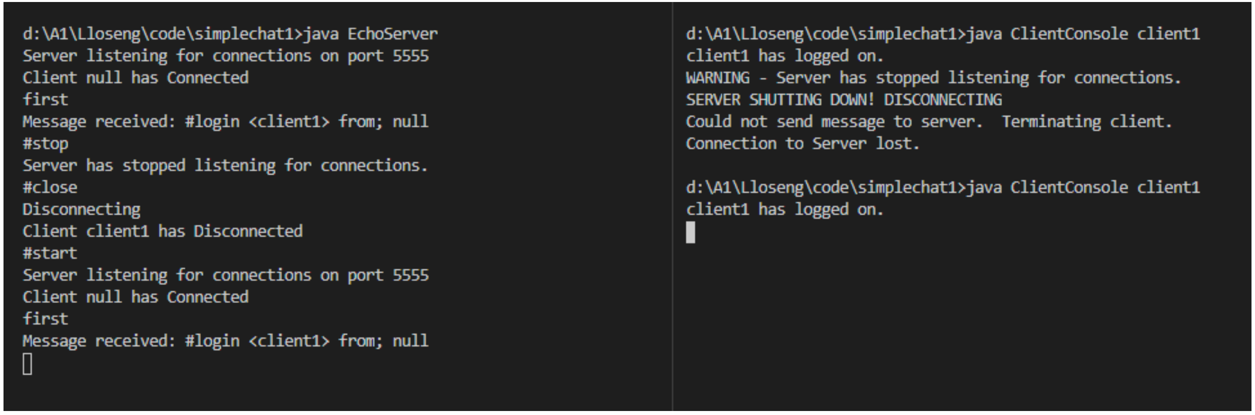
Expected result:
1. The server closes, restarts and then displays:

Server listening for connections on port 5555.

2. The client connects normally as described in Testcase 2004.

Cleanup: (Unless proceeding to Testcase 2010)
Hit CTRL+C to kill the client.
Type #quit to kill the server.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
first
Message received: #login <client1> from; null
#stop
Server has stopped listening for connections.
#close
Disconnecting
Client client1 has Disconnected
#start
Server listening for connections on port 5555
Client null has Connected
first
Message received: #login <client1> from; null
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
WARNING - Server has stopped listening for connections.
SERVER SHUTTING DOWN! DISCONNECTING
Could not send message to server.  Terminating client.
Connection to Server lost.

d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
```

<hr>
Testcase 2010: PASSED
System: Simple Chat   Phase: 2 and subsequent
Client termination command check
Severity: 1

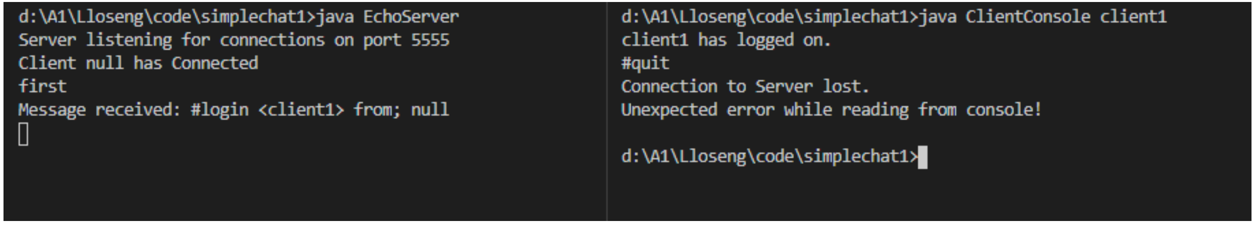
Instructions:
1. Start a client (Testcase 2002, instruction 1).

2. Type "#quit" into the client's console.

Expected result:
1. Client terminates.

Cleanup: (If client is still active)
Hit CTRL+C to kill the client.

Console Logs



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
first
Message received: #login <client1> from; null
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
#quit
Connection to Server lost.
Unexpected error while reading from console!

d:\A1\Lloseng\code\simplechat1>
```

<hr>
Testcase 2011: PASSED
System: Simple Chat   Phase: 2 and subsequent
Client logoff check
Severity: 1

Instructions:
1. Start a server (Testcase 1001, instruction 1), and then connect a single local client to this server.

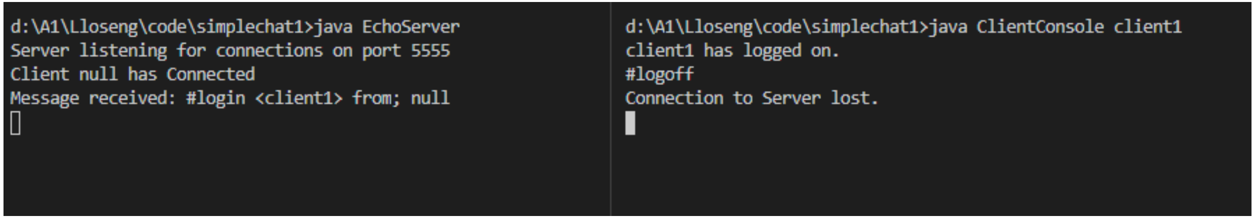
2. Type "#logoff" into this client's console.

Expected results:
1. Client disconnects and displays

Connection closed. (Under NT, it will display
Abnormal termination of connection.)

Cleanup: (Unless proceeding to Testcase 2012)
Type "#quit" to kill the client.

Console Logs



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client1> from; null
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
#logoff
Connection to Server lost.
```

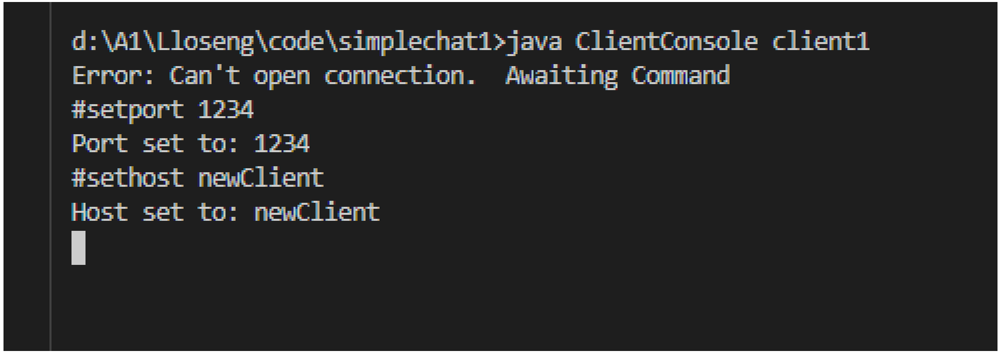
<hr>
Testcase 2012: PASSED
System: Simple Chat   Phase: 2 and subsequent
Client host and port setup commands check
Severity: 1

Instructions:
1. Start a client but no servers, and attempt to connect using default arguments.

2. At the client's console, type
"#sethost &lt;newhost&gt;"
where &lt;newhost&gt; is the name of a computer on the network

```
3. At the client's console, type
"#setport 1234".

Expected result:
1. The client displays

Host set to: &lt;newhost&gt;
port set to: 1234.

Cleanup:
Type #quit to kill the client.
```

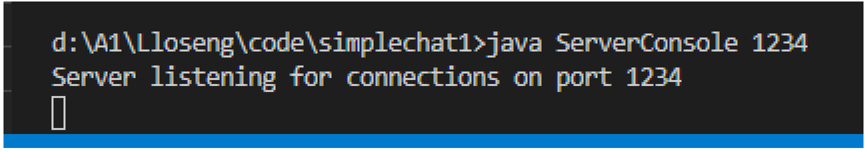
Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
Error: Can't open connection.  Awaiting Command
#setport 1234
Port set to: 1234
#sethost newClient
Host set to: newClient
```

```
<hr>
Testcase 2013: PASSED
System: Simple Chat                Phase: 2 and subsequent
Starting a server on a non-default port
Severity: 1

Instructions:
1. Start a server by typing java ServerConsole 1234.

Expected result:
1. The server displays

Server listening for connections on port 1234.

Cleanup (Unless proceeding to Testcase 2014)
Type #quit to kill the server.
```

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java ServerConsole 1234
Server listening for connections on port 1234
```

<hr>
Testcase 2014: FAILED
System: Simple Chat   Phase: 2 and subsequent
Connecting a client to a non-default host or port
Severity: 1

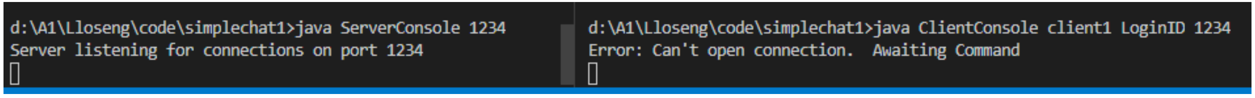
Instructions:
1. Start a server on port 1234 (Testcase 2013)

2. On a different computer, start a client by typing
"java ClientConsole &lt;loginID&gt; &lt;host&gt; 1234"
replacing the parameters by appropriate values.

Expected Result:

1. The connection occurs normally.

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java ServerConsole 1234
Server listening for connections on port 1234
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1 LoginID 1234
Error: Can't open connection.  Awaiting Command
```

Comments:
As mentioned in the “what doesn’t work” section, the port selection via parameters seem to be broken.

<hr>
Testcase 2015: PASSED
System: Simple Chat   Phase: 2
Multiple remote clients disconnections and reconnections
Severity: 1

Instructions: PASSED
1. Start a server (Testcase 2001, instruction 1).

2. On different computers, start clients (1 or 2 per computer) and connect them to the server.

3. Exchange data.

4. Close the server using the #close command.

5. Change the server's port by typing
"#setport &lt;newport&gt;"

6. Restart the server using the #start command.

7. Change the ports of each clients using the #setport command.

8. Reconnect the clients to the server by using the #login &lt;loginID&gt; command.

9. close the server by using the #quit command.

Expected results:
1. The first set of connections occur normally.

2. When the server is closed, all clients are disconnected.

3. The server displays the following message when the #setport command is used:

port set to: &lt;newport&gt;.

4. The server restarts and displays:

Server listening for connections on port &lt;newport&gt;.

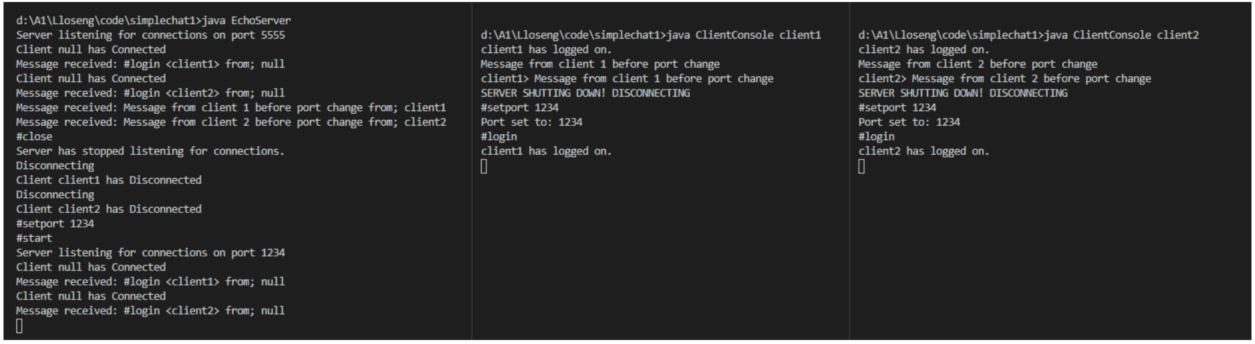
5. The clients change port as in Testcase 2012.

6. The clients reconnect normally.

7. The clients are disconnected when the server quits.

Cleanup:
Type #quit to kill the clients
Type #quit to kill the server (if still active)

Console Logs:



```
d:\A1\Lloseng\code\simplechat1>java EchoServer
Server listening for connections on port 5555
Client null has Connected
Message received: #login <client1> from; null
Client null has Connected
Message received: #login <client2> from; null
Message received: Message from client 1 before port change from; client1
Message received: Message from client 2 before port change from; client2
#close
Server has stopped listening for connections.
Disconnecting
Client client1 has Disconnected
Disconnecting
Client client2 has Disconnected
#setport 1234
#start
Server listening for connections on port 1234
Client null has Connected
Message received: #login <client1> from; null
Client null has Connected
Message received: #login <client2> from; null
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client1
client1 has logged on.
Message from client 1 before port change
client1> Message from client 1 before port change
SERVER SHUTTING DOWN! DISCONNECTING
#setport 1234
Port set to: 1234
#login
client1 has logged on.
```

```
d:\A1\Lloseng\code\simplechat1>java ClientConsole client2
client2 has logged on.
Message from client 2 before port change
client2> Message from client 2 before port change
SERVER SHUTTING DOWN! DISCONNECTING
#setport 1234
Port set to: 1234
#login
client2 has logged on.
```

<hr>
Testcase 2016
System: Simple Chat   Phase: 2 and subsequent
Client changing hosts
Severity: 1

Instructions: Failed

1. On two different computers, start servers on the default port.

2. On a third computer, start a client and connect it to one of the two servers.

3. Logoff from that server using the #logoff command.

4. Change the host name by using the #sethost &lt;otherhost&gt; where &lt;otherhost&gt; is the name of the other computer on the network

5. Log on to this new host using the #login &lt;loginID&gt; command.

Expected results:

1. The two servers start up normally.

2. The client connects to the first server normally.

3. When the client disconnects it displays

Connection closed. (Exception: NT will display Abnormal termination of connection)

4. When the client disconnects, the server displays:

&lt;loginID&gt; has disconnected.

5. The client changes host as in Testcase 2012.

6. The client reconnects normally as in Testcase 2015.

Cleanup (Unless proceeding to Testcase 2017):
Type #quit to kill the servers (if proceeding, kill only the one with no connected clients)
Type #quit to kill the client

Comments:
Unable to get test cases working on local network. Not exactly sure why.

<hr>
Testcase 2017
System: Simple Chat Phase: 2 and subsequent
Client quitting or logging off a server with multiple connections
Severity: 1

Instructions:

```
1. Start a server and connect multiple (at least 3) clients.

2. In one client's console, type "#quit".

3. In a second client's console type "#logoff".

Expected results:
1. In both cases, all remaining clients and the server get the following
message:

&lt;loginID&gt; has disconnected.

2. The clients display:

Connection Closed. (Exception: NT will display "Abnormal termination of
connection." when the logoff command is used.)

<hr>
Testcase 2018
System: Simple Chat                Phase: 2
Different platform tests
Severity: 1

Instructions:
1. Repeat all these tests on different platforms (95/98, NT, UNIX).

Expected results:
1. The same as before.

<hr>
Testcase 2019
System: Simple Chat                Phase: 2
Interaction between different platforms
Severity: 1

Instructions:
1. Repeat Testcases 2005, 2009, 2010-2019 mixing the platforms involved.

Expected results:
1. The same as before.

</pre>
</body>
</html>
```